Git Commands

1. git init (to initialize an empty repository)
2. git add filename (adds file to staging area)
3. git add .(adds all local flies to staging area)
4. git commit -m “message” (commit the files with messages)
5. git status (return name of branch & if anything to commit or not)
6. git config provides the configuration files. Configurations can be set globally or can be limited to repository.

For globally : git config –global user.name “name”

git config –global user.email “emailid”

For only repository: git config user.name “name”

git config user.email “emailid”

1. git branch (give the name of all branches)
2. git branch “branch name” (create a new branch)
3. git checkout nameofthebranch (switched to the branch)
4. git merge branchname (to merge the file in the required branch)
5. git branch-d branchName( checks to see if anything needs to merge or not before deleting the branch)

Alternative

Git branch -D branchName ( deletes the branch without checking if anything to merge or not)

1. // delete branch remotely

git push origin --delete remoteBranchName

Advanced Commands

git checkout -b “branchname” (creates a branch as well as switch to the branch)

Connecting with remote repository

1. git remote add origin sshlink of the github account (add origin)
2. git clone sshlink of the github account (it will pull all the files from remote repository)
3. git pull origin nameofbranch (to pull all the updated code of the current branch from remote to local repository)
4. git push origin branchname (to push the file to the remote repository from local repository)

Note: Before pushing to remote repository one needs to add and commit to staging )

Advanced commands

1. git stash -u (puts all the uncommitted code to stash and gives a clean directory);
2. git stash list (gives the modification list of the stash)
3. git stash show ( gives all the file names of stash)
4. git log (gives the log info of the current commit activity)
5. git log – author=”nameofauthor” (gives the log activity of that particular author)
6. git log –before=”specifydate” (gives the log activity of the mentioned before date)
7. git revert hashcode (revert back to previous commit). [hascode can be generated by git log—oneline]
8. git revert HEAD ( takes the last commit)
9. git rebase branchName (takes a set of commits ,copies them and store them outside repository)s